**Project**

**on**

**Library Management System Documentation**

**Project Overview**

The Library Management System is an online tool designed to help a typical college library manage book issuance and retrieval, maintain book and user records, and calculate fines. Developed using SQL database, this system provides a user-friendly interface for both students and teachers to interact with the library's resources.

**Project Objectives**

* **Streamline Book Issuance and Returns**: Simplify the process of issuing and returning books.
* **Manage User Records**: Maintain records for both teachers and students.
* **Track Book Availability**: Ensure that the library inventory is accurately tracked.
* **Calculate Fines**: Automatically calculate fines for overdue books.

**Database Design**

**Tables**

1. **Books**
   * **BookID**: INT (Primary Key)
   * **Title**: VARCHAR(255)
   * **Author**: VARCHAR(255)
   * **Category**: VARCHAR(100)
   * **UniqueID**: VARCHAR(100) (Unique)
2. **Users**
   * **UserID**: INT (Primary Key)
   * **UserName**: VARCHAR(255)
   * **UserType**: VARCHAR(50) ('Student' or 'Teacher')
3. **BookIssues**
   * **IssueID**: INT (Primary Key)
   * **BookID**: INT (Foreign Key)
   * **UserID**: INT (Foreign Key)
   * **IssueDate**: DATE
   * **ReturnDate**: DATE
   * **Fine**: DECIMAL(10, 2)

**ER Diagram**

**SQL Queries**

**Create Database**

CREATE DATABASE LibraryDB;

USE LibraryDB;

**Create Tables**

**Books Table**

CREATE TABLE Books (

BookID INT AUTO\_INCREMENT PRIMARY KEY,

Title VARCHAR(255) NOT NULL,

Author VARCHAR(255) NOT NULL,

Category VARCHAR(100) NOT NULL,

UniqueID VARCHAR(100) NOT NULL UNIQUE

);

**Users Table**

CREATE TABLE Users (

UserID INT AUTO\_INCREMENT PRIMARY KEY,

UserName VARCHAR(255) NOT NULL,

UserType VARCHAR(50) NOT NULL

);

**BookIssues Table**

CREATE TABLE BookIssues (

IssueID INT AUTO\_INCREMENT PRIMARY KEY,

BookID INT,

UserID INT,

IssueDate DATE NOT NULL,

ReturnDate DATE NOT NULL,

Fine DECIMAL(10, 2),

FOREIGN KEY (BookID) REFERENCES Books(BookID),

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

**Insert Data**

**Insert Books**

INSERT INTO Books (Title, Author, Category, UniqueID) VALUES

('Introduction to Algorithms', 'Thomas H. Cormen', 'Computer Science', 'CS101'),

('Data Structures', 'Seymour Lipschutz', 'Computer Science', 'CS102'),

('Artificial Intelligence', 'Stuart Russell', 'Computer Science', 'CS103');

**Insert Users**

INSERT INTO Users (UserName, UserType) VALUES

('Aravind', 'Student'),

('Sushma', 'Teacher');

**Issue and Return Books**

**Issue a Book**

INSERT INTO BookIssues (BookID, UserID, IssueDate, ReturnDate, Fine) VALUES

(1, 1, '2024-08-01', '2024-08-21', 0.00),

(2, 2, '2024-08-01', '2024-09-01', 0.00);

**Return a Book and Calculate Fine**

UPDATE BookIssues

SET ReturnDate = '2024-08-25', Fine = 50.00

WHERE IssueID = 1;

**Retrieve Data**

**Retrieve All Books in a Category**

SELECT \* FROM Books

WHERE Category = 'Computer Science';

**Retrieve Issued Books**

SELECT b.Title, u.UserName, bi.IssueDate, bi.ReturnDate, bi.Fine

FROM BookIssues bi

JOIN Books b ON bi.BookID = b.BookID

JOIN Users u ON bi.UserID = u.UserID;

**Retrieve Books Issued by a Specific User**

SELECT b.Title, bi.IssueDate, bi.ReturnDate, bi.Fine

FROM BookIssues bi

JOIN Books b ON bi.BookID = b.BookID

WHERE bi.UserID = 1;

**SQL Server**

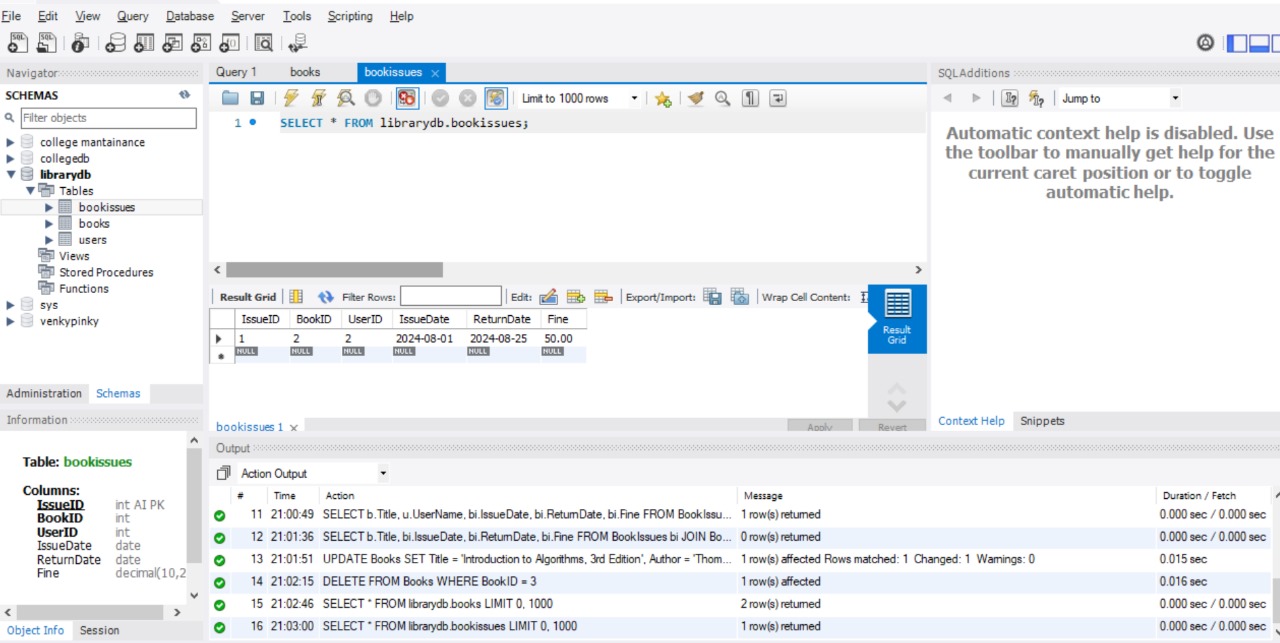
**Using SQL Server Management Studio (SSMS)**

1. **Open SQL Server Management Studio (SSMS)**.
2. **Connect to your database server**.
3. **Right-click on your database (LibraryDB)** and go to **Tasks > Generate Scripts**.
4. **Follow the wizard** to save the script to your desktop.

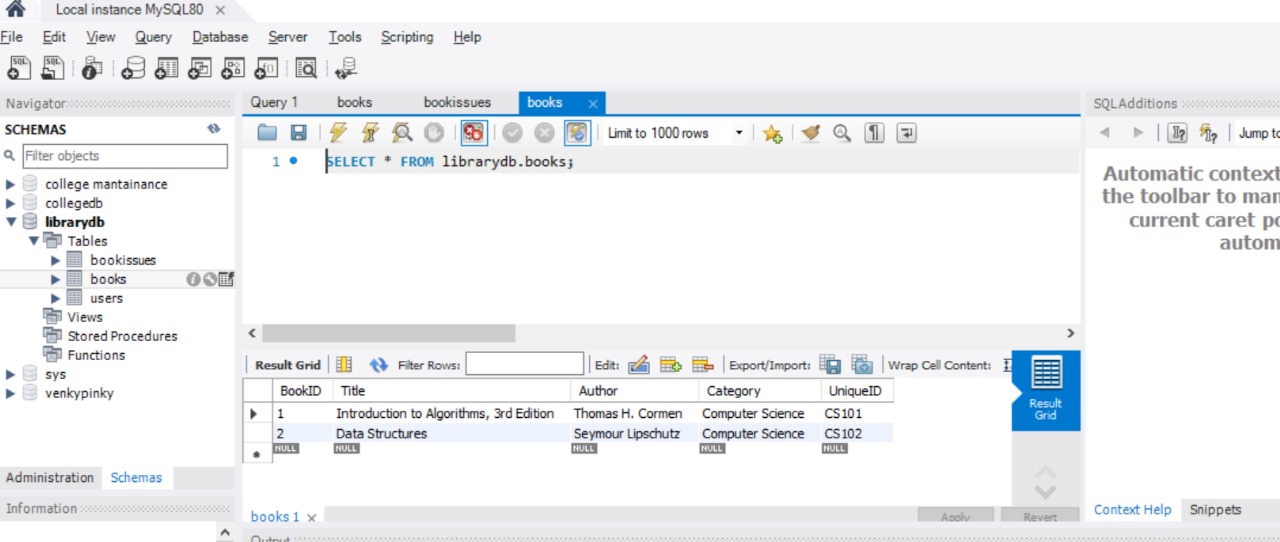
**Conclusion**

This documentation provides a comprehensive overview of the Library Management System, including database design, SQL queries for creating and managing data, and steps for exporting the database. By following this guide, the library can effectively manage book issuance, user records, and fines.

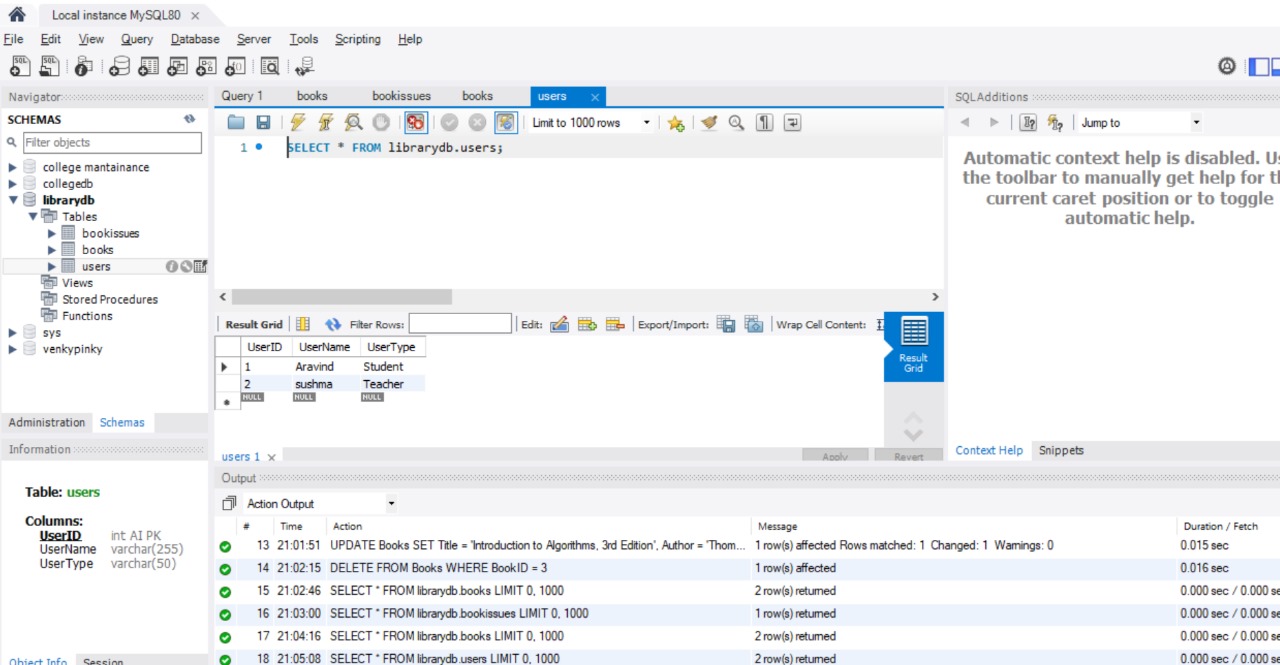
**Output**



**Book issuses**



**Books**



**users**